CI/CD

GIVE YOUR APPLICATION AUTO-DEPLOY SUPERPOWERS

INTRODUCTION

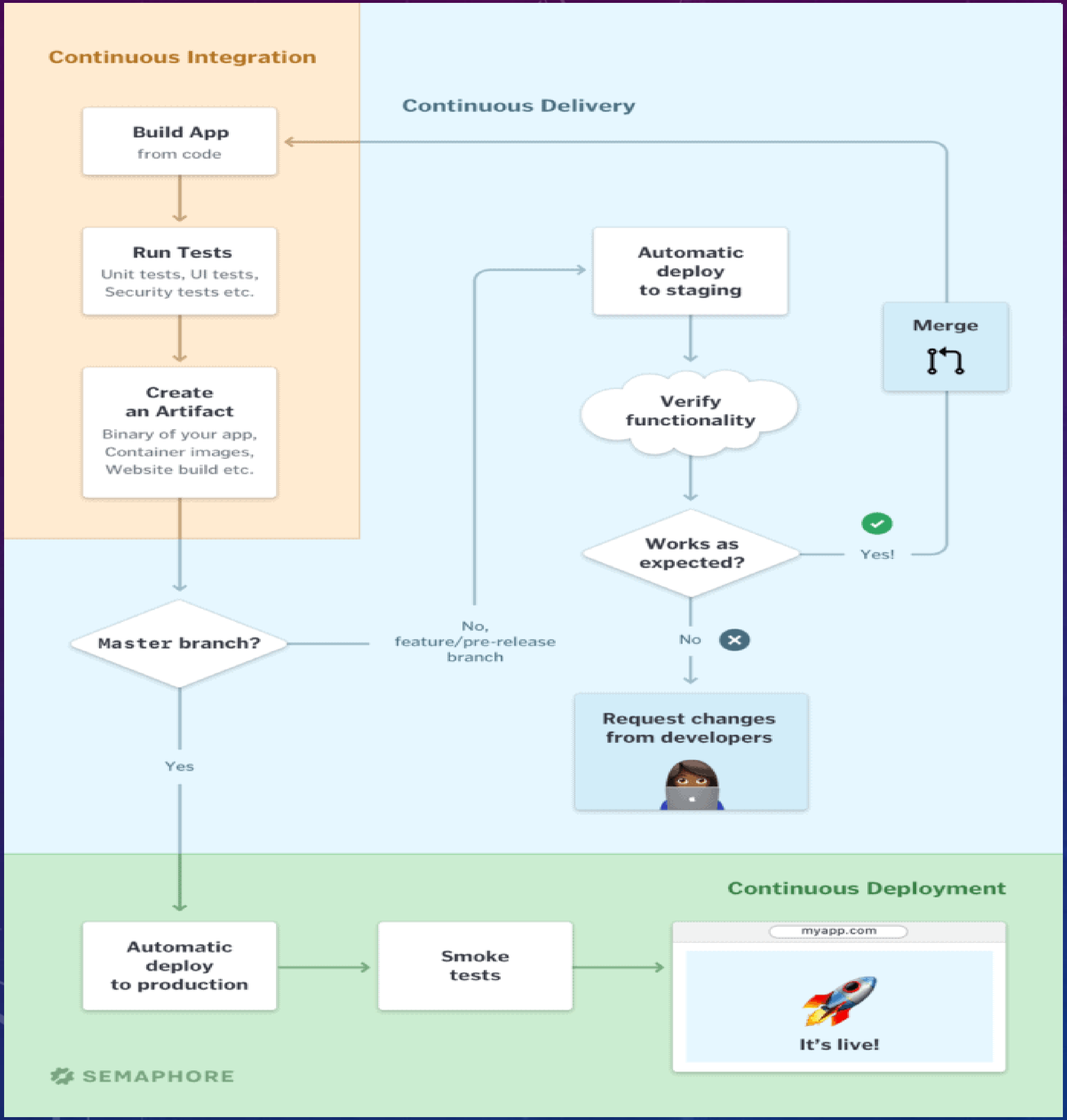
- CI/CD is a way of developing software in which you're able to release updates at any time in a sustainable way. When changing code is routine, development cycles are more frequent, meaningful and faster.
- "CI/CD" stands for the combined practices of Continuous Integration (CI) and Continuous Delivery (CD).
- Continuous Integration is a prerequisite for CI/CD, and requires:
- Developers to merge their changes to the main code branch many times per day.
- Each code merge to trigger an automated code build and test sequence. Developers ideally receive results in less than 10 minutes, so that they can stay focused on their work.

WHY ARE CI/CD SO IMPORTANT?

- Some of the benefits of CI/CD tools include:
- Beat overwhelm by building, testing, deploying, and monitoring your code from a single CI/CD tool.
- Automate repetitive tasks when building, testing, deploying, and maintaining software.
- Speed up time-to-market by minimizing manual labor and risk of human error.
- Free up software engineers to focus on more creative tasks, such as inventing new features, rather than doing repetitive work.
- Configure the right environment for software or code to run at its best.
- Identify and resolve issues before they affect customers.
- Support seamless collaboration between distributed software engineers, such as creating tasks, backlogs, and keeping track of CI/CD processes.
- Test code changes to ensure that only high-quality, secure code is pushed to live environments.
- Some CI/CD tools serve as version control, knowledge management, or cost management tools.

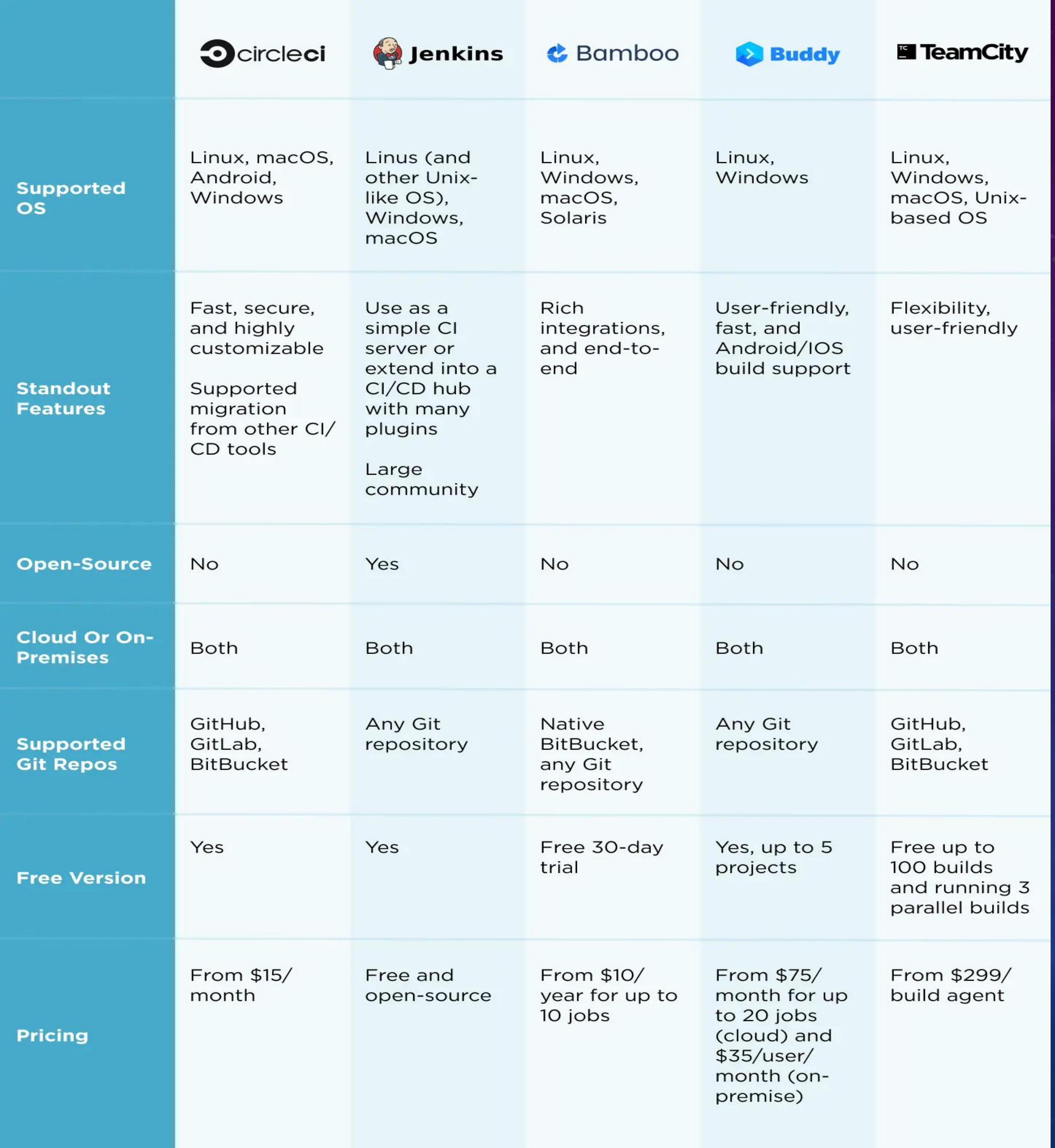
CI/CD PRINCIPLES

- Continuous Delivery practices take CI further by describing principles for successful production deployments:
- **Architect the system in a way that supports iterative releases**. Avoid tight coupling between components. Implement metrics that help detect issues in real-time.
- **Practice test-driven development to always keep the code in a deployable state**. Maintain a comprehensive and healthy automated test suite. Build in monitoring, logging, and fault-tolerance by design.
- **Work in small iterations**. For example, if you develop in feature branches, they should live no longer than a day. When you need more time to develop new features, use feature flags.
- Developers can **push the code into production-like staging environments**. This ensures that the new version of the software will work when it gets in the hands of users.
- **Anyone can deploy any version** of the software to any environment on demand, **at a push of a button**. If you need to consult a wiki on how to deploy, it’s game over.
- **If you build it, you run it**. Autonomous engineering teams should be responsible for the quality and stability of the software they build. This breaks down the silos between traditional developers and operations groups, as they work together to achieve high-level goals.
- To make CI/CD a reality, you need to automate everything that you can in the software delivery process and run it in a CI/CD pipeline

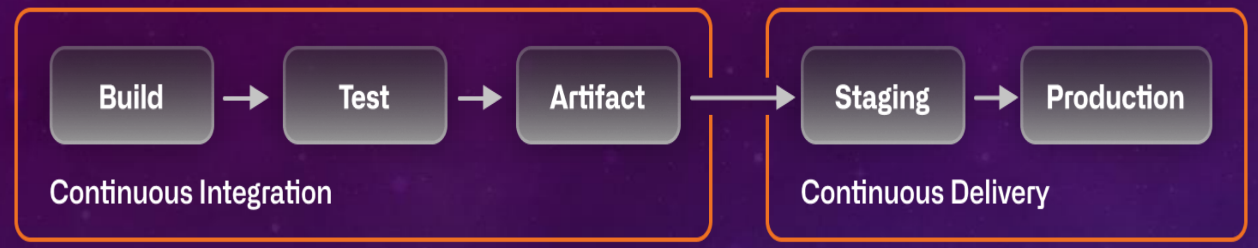

Continuous Integration
Build App
from code
Run Tests
Unit tests, UI tests, Security tests etc.
Create an Artifact
Binary of your app, Container images, Website build etc.
Continuous Delivery
Automatic deploy to staging
Verify functionality
Works as expected?
Yes!
Merge
No
Request changes from developers
Master branch?
No, feature/pre-release branch
Yes
Continuous Deployment
Automatic deploy to production
Smoke tests
myapp.com
It's live!
SEMAPHORE

CI/CD pipeline

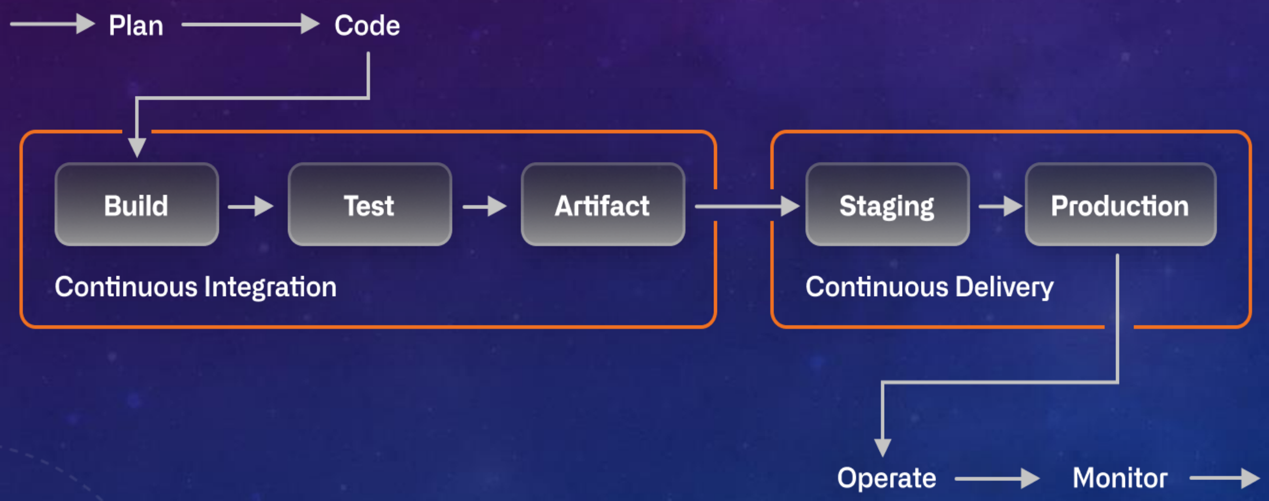
END-TO-END CI/CD TOOLS



	circleci	Jenkins	Bamboo	Buddy	TeamCity
Supported OS	Linux, macOS, Android, Windows	Linus (and other Unix-like OS), Windows, macOS	Linux, Windows, macOS, Solaris	Linux, Windows	Linux, Windows, macOS, Unix-based OS
Standout Features	Fast, secure, and highly customizable Supported migration from other CI/CD tools	Use as a simple CI server or extend into a CI/CD hub with many plugins Large community	Rich integrations, and end-to-end	User-friendly, fast, and Android/IOS build support	Flexibility, user-friendly
Open-Source	No	Yes	No	No	No
Cloud Or On-Premises	Both	Both	Both	Both	Both
Supported Git Repos	GitHub, GitLab, BitBucket	Any Git repository	Native BitBucket, any Git repository	Any Git repository	GitHub, GitLab, BitBucket
Free Version	Yes	Yes	Free 30-day trial	Yes, up to 5 projects	Free up to 100 builds and running 3 parallel builds
Pricing	From $15/month	Free and open-source	From $10/year for up to 10 jobs	From $75/month for up to 20 jobs (cloud) and $35/user/month (on-premise)	From $299/build agent

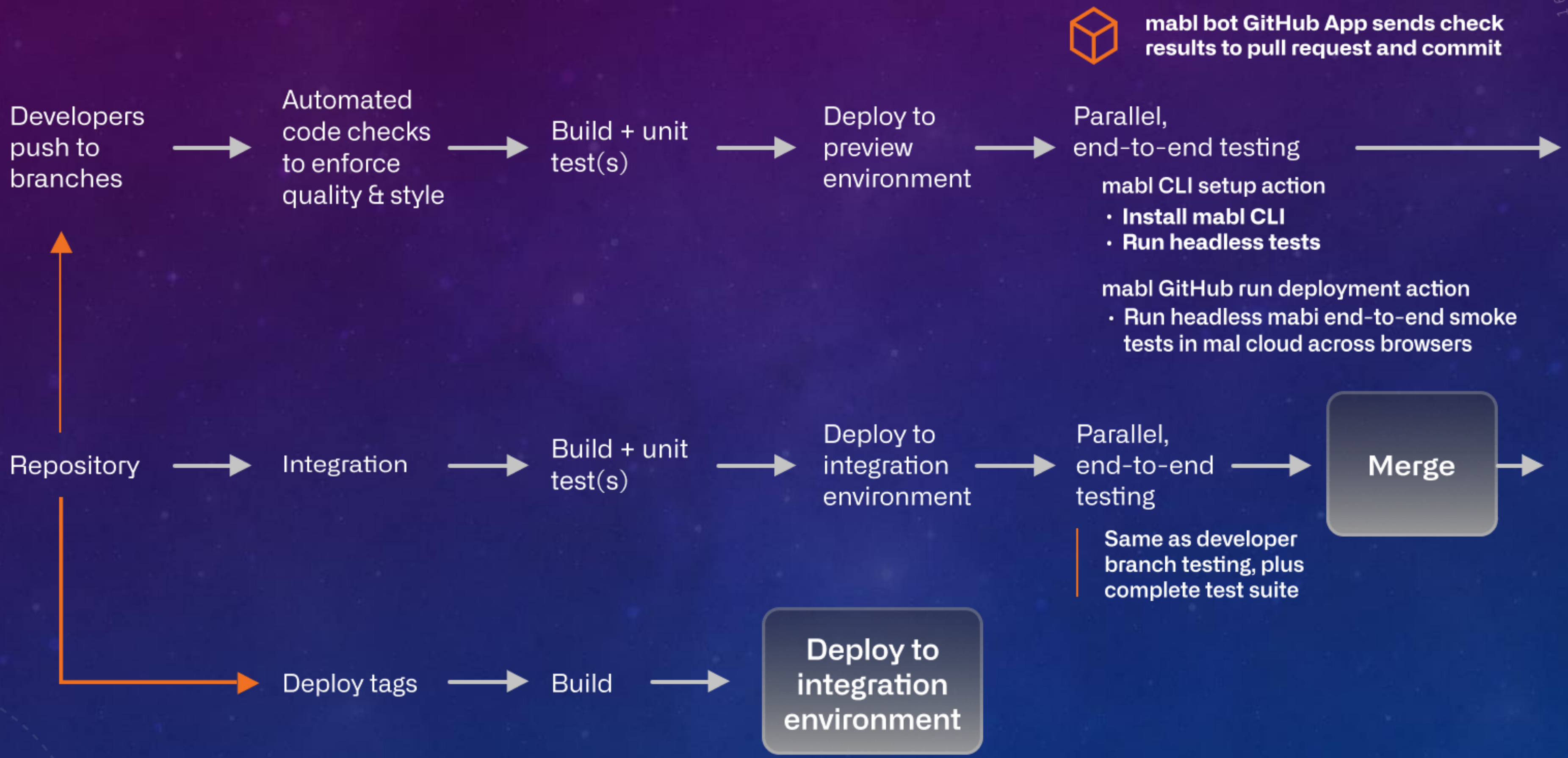


A CI/CD PIPELINE



Continuous delivery vs. continuous deployment

EXAMPLE CI/CD WORKFLOW



Developers push to branches
Automated code checks to enforce quality & style
Build + unit test(s)
Deploy to preview environment
Parallel, end-to-end testing
Repository
Integration
Build + unit test(s)
Deploy to integration environment
Parallel, end-to-end testing
Merge
Deploy tags
Build
Deploy to integration environment

mabl bot GitHub App sends check results to pull request and commit

mabl CLI setup action

- Install mabl CLI
- Run headless tests

mabl GitHub run deployment action

- Run headless mabi end-to-end smoke tests in mal cloud across browsers

Same as developer branch testing, plus complete test suite

REDUCE COSTS AND BOOST PROFITS

- CI/CD is also good for the bottom line. It standardizes deployment processes across all projects, and, done right, it enables teams to systematically test every change made to the source code.
- As a result, this process stands to dramatically reduce the likelihood that any bugs or errors slip through the cracks and cause problems down the line. Done right, this practice can lower development costs by eliminating many of the costs incurred while building and testing code changes.
- Teams spend less time on testing and bug fixes, meaning organizations spend less money on tasks that don't provide any value to the business or its customers.
- And because CI/CD also makes it easier to deliver high-quality products to market faster and respond to feedback as it comes in, organizations stand to see an increase in profits. Customers stick around longer and will likely recommend your products to others in their network.

Thank You

Khairy Ibrahim Mohamed

Eng.khairyibrahem@gmail.com